

Estruturas de Repetição

PARTE 1:

Repetições
Contadores

Sumário

- Estrutura de Repetição
- Contadores

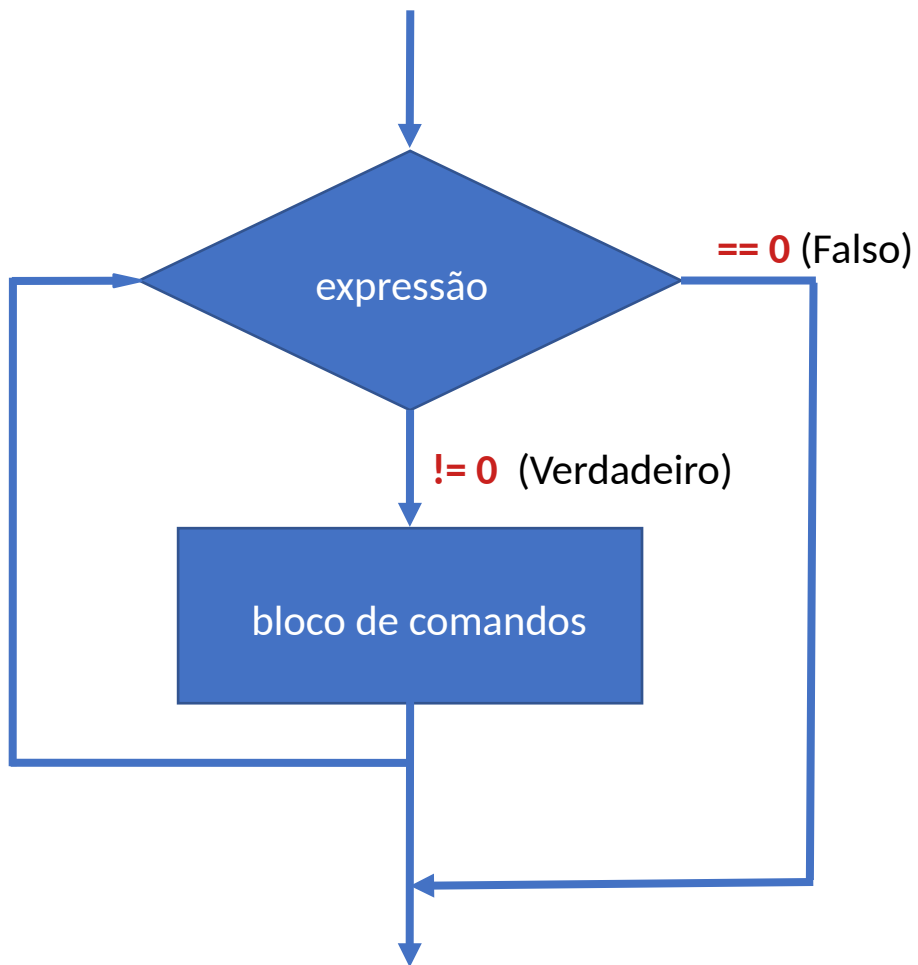
Repetições – Primeiro exemplo

Escrever “Mãe, eu te amo!” 10 vezes

```
/* Programa 'Mãe eu te amo' */
#include <iostream>
using namespace std;
int main()
{
    cout << “Mãe, eu te amo!” << endl;
    cout << “Mãe, eu te amo!” << endl;
    cout << “Mãe, eu te amo!” << endl;
    cout << “Mãe, eu te amo!” << endl;
    cout << “Mãe, eu te amo!” << endl;
    cout << “Mãe, eu te amo!” << endl;
    cout << “Mãe, eu te amo!” << endl;
    cout << “Mãe, eu te amo!” << endl;
    cout << “Mãe, eu te amo!” << endl;
    cout << “Mãe, eu te amo!” << endl;
    return 0;
}
```

- O programa tem um fluxo sequencial
- O programa não explora o que o computador faz de melhor → repetir

Comando de repetição: **while**



Semântica

Enquanto o valor de **expressão** for **!= 0** (Verdadeiro), executa **bloco de comandos**.

Sintaxe em C++

```
while (expressão)  
    bloco de comandos;
```

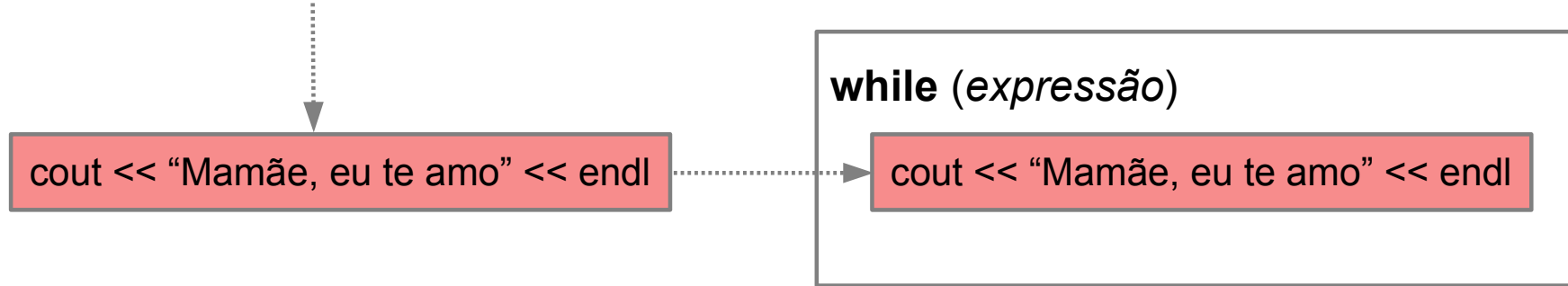
```
{  
    comando 1;  
    comando 2;  
    .....  
}
```

Desenvolvimento de um Programa com Repetição

- Não é fácil !!!
- Você não escreve o comando repetitivo de forma linear
- Passos:
 1. Reconhecer o padrão repetitivo
 2. Definir controles de repetição (nesta aula, com contador)
 3. Definir a condição de parada

Padrão repetitivo

O que você quer repetir?



Você tem que reconhecer o **Padrão Repetitivo**

Controle da repetição

Como controlar a repetição?

Inicialização:

contador começa com valor **0** (zero)
[No início, não começou a contagem e nenhuma mensagem exibida ainda]

Incremento:

a cada repetição, conta mais **1**
[Conta a mensagem exibida]

```
contador = 0;
```

```
while (expressão)
```

```
cout << "Mãe, eu te amo" << endl
```

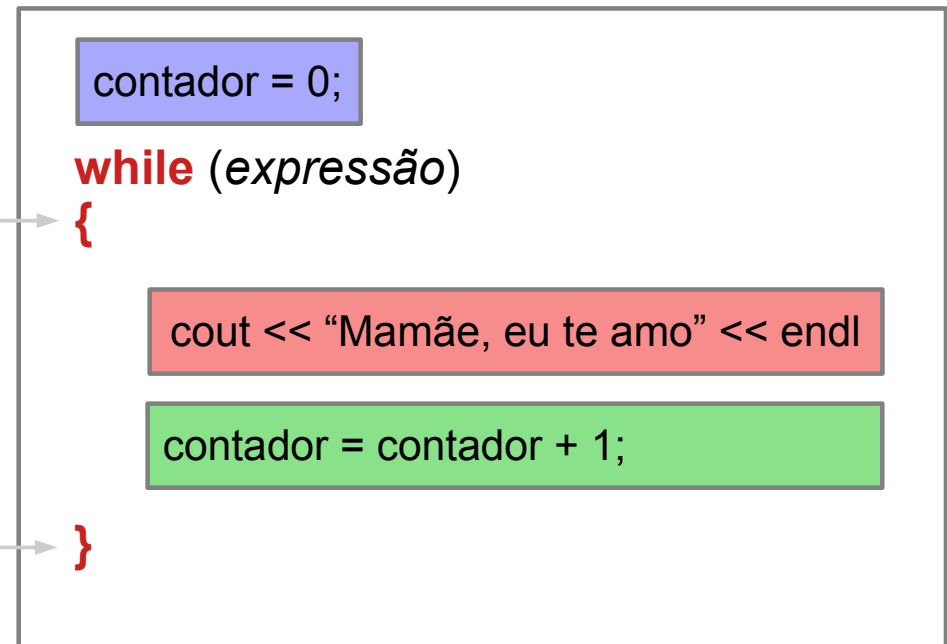
```
contador = contador + 1;
```

Bloco de repetição

O incremento **TAMBÉM** deve fazer parte da repetição:
bloco de comandos da repetição

Os comandos entre **{** e **}** criam um único **bloco de comandos**.

bloco da repetição:
conjunto de comandos
que se repetem



Condição de Parada

A repetição continua enquanto o que for verdadeiro ($\neq 0$) ?

Condição de parada:

repete-se a mensagem enquanto o contador variar de 0 a 9 (inclusive)

Terminada a repetição:

contador \rightarrow número de repetições \rightarrow 10

```
contador = 0;
```

```
while ( contador < 10 )  
{
```

```
    cout << "Mãe, eu te amo" << endl
```

```
    contador = contador + 1;
```

```
}
```

```
// O que é o valor de contador neste ponto?
```

Programa completo

```
/* Programa 'Mãe eu te amo' */

#include <iostream>
using namespace std;

int main()
{
    int contador;

    contador = 0;
    while ( contador < 10 )
    {
        cout << "Mãe, eu te amo!" << endl;
        contador = contador + 1;
    }

    return 0;
}
```

Escrever os números de 1 a 10

```
/* Programa 'escreve1a10' */
#include <iostream>
using namespace std;

int main()
{
    cout << 1 << endl;
    cout << 2 << endl;
    cout << 3 << endl;
    cout << 4 << endl;
    cout << 5 << endl;
    cout << 6 << endl;
    cout << 7 << endl;
    cout << 8 << endl;
    cout << 9 << endl;
    cout << 10 << endl;
    return 0;
}
```

Existe um padrão repetitivo, mas o valor muda em cada repetição

→ Então isto é uma

variável

```
/* Programa 'escreve1a10' */
#include <iostream>
using namespace std;

int main()
{
    int contador;

    /* inicialização */
    while ( /* expressão */ ) {
        cout << contador << endl;

        /* incremento */
    }

    return 0;
}
```

Escrever os números de 1 a 10

```
/* Programa 'escreve1a10' */

#include <iostream>
using namespace std;

int main()
{
    int contador;

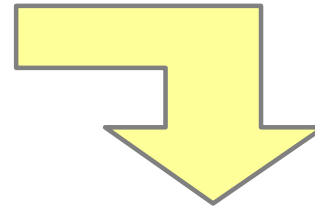
    contador = 1; // inicialização
    while ( contador <= 10 )
    {
        cout << contador << endl;
        contador = contador + 1; // incremento
    }
    return 0;
}
```



PAUSA

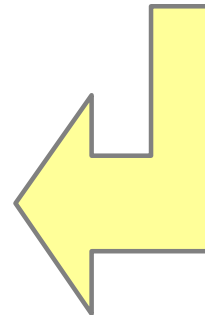
Escrever os inteiros pares no intervalo 1..100

```
/* Programa 'pares1a100' */  
#include <iostream>  
using namespace std;  
int main()  
{  
    cout << 2 << endl;  
    cout << 4 << endl;  
    cout << 6 << endl;  
    ....  
    cout << 100 << endl;  
    return 0;  
}
```



O **padrão repetitivo** é o mesmo,
mas vamos mudar
o nome da variável de contagem para **k**:

```
/* inicialização */  
while (/* expressão */) {  
    cout << k << endl;  
    /* incremento */  
}
```



Escrever os inteiros pares no intervalo 1..100

O valor inicial de **k** é 2:

```
k = 2;  
while ( /* expressão */  
{  
    cout << k << endl;  
    /* incremento */  
}
```

O incremento é de 2 em 2

```
k = 2;  
while ( /* expressão */  
{  
    cout << k << endl;  
    k = k + 2;  
}
```

Escrever os inteiros pares no intervalo 1..100

Repete enquanto valor de **k** menor ou igual a 100:

```
int k;  
k = 2;  
while ( k <= 100 ) {  
    cout << k << endl;  
    k = k + 2;  
}
```

Programa completo:

```
/* Programa 'pares1a100' */  
#include <iostream>  
using namespace std;  
int main()  
{  
    int k;  
    k = 2;  
    while ( k <= 100 ) {  
        cout << k << endl;  
        k = k + 2;  
    }  
  
    return 0;  
}
```


Exercício

Escrever um programa que leia dois valores inteiros $n1$, $n2$, onde $n1 \leq n2$, e exiba na tela os valores inteiros no intervalo $[n1, n2]$.

Qual o valor inicial de k ?
Qual a condição de parada?

Escrever os números de **n1** a **n2**

```
/* Programa 'escreve_N1aN2' */

#include <iostream>
using namespace std;

int main()
{
    int n1, n2, k;

    cin >> n1 >> n2;    // Leitura dos extremos do intervalo

    k = n1;              // Inicialização de k ← extremo inferior

    while ( k <= n2 )   // Condição de parada → k atinge extremo superior
    {
        cout << k << endl;
        k = k + 1;      // Incremento de k
    }

    return 0;
}
```

Escrever valores de uma série

1^2 2^2 3^2 $4^2 \dots$ 100^2

```
/* Programa 'seriePotencias' */  
#include <iostream>  
using namespace std;  
int main()  
{  
    cout << 1 * 1 << endl;  
    cout << 2 * 2 << endl;  
    cout << 3 * 3 << endl;  
    cout << 4 * 4 << endl;  
    ....  
    cout << 100 * 100 << endl;  
    return 0;  
}
```

padrão repetitivo

```
while ( ..... )  
{  
    cout << k * k << endl;  
    /* incremento */  
}
```

Escrever o programa completo.

Escrever valores de uma série

```
/* Programa 'seriePotencias' */  
  
#include <iostream>  
using namespace std;  
  
int main()  
{  
    int k;  
    k = 1;           // Inicialização de  $k \leftarrow 1$   
    while ( k <= 100 ) // Condição de parada  $\rightarrow k > 100$   
    {  
        cout << k * k << endl; // Imprime o quadrado de  $k$   
        k = k + 1;           // Avança para o próximo inteiro  
    }  
    return 0;  
}
```

Calcular a soma de pares de números

Ler uma sequência de 20 pares de números e, para cada par, escrever sua soma.

Exemplo:

$$\mathbf{5\ 7}$$
$$5+7=12$$

$$\mathbf{9\ 5}$$
$$9+5=14$$

...

$$\mathbf{13\ 13}$$
$$13+13=26$$

Programa sequencial 'somaParesValores'

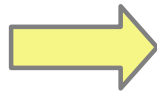
```
/* Programa 'somaParesValores' */
#include <iostream>
using namespace std;

int main()
{
    int n1, n2;

    cin >> n1 >> n2 ;
    cout << n1 << " + " << n2 << " = " << n1 + n2 << endl;
    cin >> n1 >> n2 ;
    cout << n1 << " + " << n2 << " = " << n1 + n2 << endl;
    ...
    cin >> n1 >> n2 ;
    cout << n1 << " + " << n2 << " = " << n1 + n2 << endl;
    return 0;
}
```

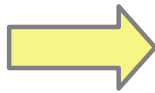
Programa 'somaParesValores2'

Padrão repetitivo



```
cin >> n1 >> n2 ;  
cout << n1 << " + " << n2 << " = " << n1 + n2 << endl;
```

Controle da
repetição



```
contador = 0;  
while ( ... ) {  
    cin >> n1 >> n2 ;  
    cout << n1 << " + " << n2 << " = " << n1 + n2 << endl;  
    contador = contador + 1;  
}
```

Teste de parada



```
contador = 0;  
while ( contador < 20 ) {  
    cin >> n1 >> n2 ;  
    cout << n1 << " + " << n2 << " = " << n1 + n2 << endl;  
    contador = contador + 1;  
}
```

Programa 'somaParesValores2' completo

```
/* Programa 'somaParesValores' */
#include <iostream>
using namespace std;

int main()
{
    int n1, n2, contador;

    contador = 0;

    while ( contador < 20 ) {
        cin >> n1 >> n2 ;
        cout << n1 << " + " << n2 << " = " << n1 + n2 << endl;
        contador = contador + 1;
    }

    return 0;
}
```


Exercícios para aula *online*

Após assistir todas as vídeo-aulas da semana, procure trabalhar na **Lista de exercícios** do Tópico **Estruturas de Repetição**, na sala virtual da disciplina na UFPR Virtual.

Estes exercícios serão usados nas aulas *online* para esclarecer e consolidar os conceitos abordados até aqui.

Leitura complementar

Acesse as **Leituras complementares** do Tópico **Estruturas de Repetição**, na sala virtual da disciplina da UFPR Virtual.

Elas são importantes e auxiliam na compreensão dos temas abordados até aqui.

Créditos: O conteúdo original deste documento é de autoria da Profª Carmem Satie Hara (DINF/ET), e foi adaptado pelo Prof. Armando L.N. Delgado (DINF/ET) para uso na disciplina *Programação de Computadores* (CI208, CI180, CI183)

Compartilhe este documento de acordo com a licença abaixo



Este documento está licenciado com uma Licença *Creative Commons Atribuição-NãoComercial-SemDerivações* 4.0 Internacional.
<https://creativecommons.org/licenses/by-nc-sa/4.0/>